To: Distribution

From: Tom Van Vleck, Andre Bensoussan

Date: February 28, 1975

Subject: New Storage System Disk Usage

This memorandum describes some of the details of implementation of the new Multics storage system which were left unspecified in MTB-110.

DISK ADDRESSING

In order to accomodate much larger storage system device configurations, the way in which the supervisor and BOS address secondary storage must be expanded. More bits will be used in the secondary storage address, so that the total system capacity is larger; at the same time, the addressing will be reorganized to make it easier to dismount a single disk pack.

Definitions

There has been some confusion in terms between physical and logical storage-allocation entities.

RECORD

A record is a 1024-word logical unit of storage allocation. A segment is stored on secondary storage in from zero to 256 records.

SECTOR

A sector is a 64-word physical storage unit on a DSS-191 or similar disk pack. The new storage system does not require that all volumes have the same size sectors, or the same number of sectors.

MAPPING BETWEEN RECORDS AND SECTORS

The model number specified for a disk subsystem is looked up in a system table to determine the maximum capacity of the device and the constants needed to map record addresses into sector numbers. Not all sectors are part of some record: for example, in the DSS-191, eight sectors per cylinder are unused by the current disk DIM. The waste of 1% of the disk capacity is preferable to allowing access to some pages in the system's storage to require a disk arm motion during each read or write.

Addresses

The type of address used by the supervisor and by BOS must be changed in the new system.

CURRENT ADDRESSES

In the current system, BOS addresses secondary storage by specifying

device ID
area number
sector number

Area numbers start at zero. Sector numbers start at zero for each area.

The supervisor currently addresses secondary storage by specifying

device ID
record number

where record number starts at zero for the first record in each device ID and goes up by one to the maximum capacity indicated in

the FSDCT.

A "device ID" in the current system is a number which indicates which storage subsystem contains a record. The four-bit number actually specifies only a maximum of seven devices, because the high-order bit is used to indicate whether the address is on the paging device. The current values of device ID are:

0	Null Address
1	Bulk Store
2	DSS-191
3	DSS-191
4	DSU-190
5	DSU-190
6	DSU-181
7	unused

The disk strategy code uses device ID as an index into tables to determine physical parameters such as number of cylinders and maximum capacity, as well as using the three low bits of the device ID as additional address bits.

NEW ADDRESSES

All addresses in the new storage system will have the uniform format of

physical volume identifier
record number

Record numbers will begin with 0 for each physical volume. The physical volume will be specified by index in the Physical Volume Table (PVT) within the supervisor and in most parts of BOS; but when the system reports device errors, and when the operator specifies addresses to the system or punches configuration cards, he will use the name of the physical volume, which is a character string.

Multics addresses will look like this:

The indicator "S" is used as a special flag showing that the secondary storage address should be released at deactivation time. This address format allows for a maximum of 131,071

records per physical volume. (Current DSS-191's have 19,270.) If the physical capacity of a single disk drive ever exceeds 131K records (i.e. over 500 million characters per pack) then the storage system will have to treat a single pack as two or more logical packs, or some other strategy.

SPECIAL PARTITIONS

The current system divides the disk storage available on the configuration into several partitions. Usually, these are

BOS	BOS commands, runcoms, and saved core
LOG	messages written over SYSERR
DUMP	address space image saved after crash
SALV	address space for salvager paging
MULT	storage space for Multics hierarchy
PAGE	paging device

Each partition is specified by a configuration card in the BOS configuration deck which lists the starting address and extent for each device id, e.g.

```
PART MULT 0 0 0 36502. 0 0 0 0 0 0 0 0
```

which says that the MULT partition occupies records 0 to 36501 on the second device (DSS-191).

In present usage, it is very rare for any partition except the MULT partition to occupy storage on more than one physical volume. (Indeed, some code may fail to work if this is tried.) It therefore seems sensible to redefine partitions so that the MULT partition is treated as a special case, and the other partitions are restricted to reside entirely on one physical volume.

PHYSICAL VOLUME LAYOUT

This section describes the format of a Multics standard storage volume for the new storage system. All volumes will have the same internal layout, except that one volume, that which contains the root directory of the hierarchy, will also have some special per-system data in it.

Label

The first record on each volume is the volume label. It contains the name of the volume and the manufacturer's serial number in character-string format, and a unique identifer assigned when the volume is registered, called the Volume Unique ID.

Each physical volume is part of some logical volume. The name and unique ID of the logical volume are also recorded in the label.

The label also contains several 52-bit times, including the time the volume was registered, the time it was last mounted, and so forth.

Any special partitions on the volume will be described in the partition array section of the label for the volume.

If the volume is the one (and only) volume which contains the root directory for the Multics hierarchy, a special flag is set in the label, and the VTOC index of the root directory is noted.

Volume Map

The volume map for each physical volume may occupy up to three records. (Current DSS-191's will need less than 602 words.) The map will consist of a string of bits, one for each record, indicating whether the record is used or not. The fixed area of the pack and the special partitions need not be described by the volume map, since space in these areas cannot be assigned by the page fault handler.

VTOC Header

The VTOC header occupies one record. It contains only a few items which are used in initializing the VTOC entry allocation mechanism, such as the count of free VTOC entries and the VTOC index of the first entry on the free chain.

Bad Record Data

Three records are allocated for information on error history and data on which addresses on the physical volume are unusable due to errors. The exact form of this information has not been specified; presumably it will consist of a journal of errors, giving the address and type of bad status encountered.

Volume Table of Contents

The VTOC itself follows the first 8 records of the volume. If n records are allocated to the fixed part of each physical volume, then the VTOC will occupy (n-8) records.

The initial implementation of the new storage system will use 256-word VTOC entries and will fix n at 1024. Therefore, each physical volume may contain up to 4064 segments.

Rest of Volume

The rest of each physical volume will consist of records which can be

* allocated to segments. These records will be pointed to by segment maps in the VTOC entries on the volume. They will be marked used in the volume map.

* free records. These records will be marked free in the volume map.

* records containing a bad sector. These records will be marked as being used in the volume map but will not be pointed to by any segment map. The bad record data in the volume header will indicate that this record is defective.

* part of a special partition. These records will be marked as used in the volume map, and will be indicated in the volume label as part of the special partition. Whether these records are pointed to by a VTOC entry or not has not been decided. (see below) By convention, the records for all special partitions will have higher addresses than any address used for segments.

CONFIGURATION DECK

Since the concept of "device ID" goes away, the CONFIG cards which describe the storage devices available to Multics must be changed radically.

Here is an example of part of a new-style CONFIG deck:

```
VOL V1 ROOT DSK 1
VOL V2 ROOT DSK 2
VOL V3 USER DSK 3
VOL PD PAGE BULK 0
PART BOS V2 19170. 100.
PART LOG V2 17744. 256.
PART DUMP V2 18000. 1170.
PART SALV V2 17232. 512.
PART PAGE PD 0 256.
PRPH DSK A 25. 191. (DIM-dependent items)
BULK 0 256. 1 2
PAGE BULK 0 256.
```

This example describes a small configuration with only three packs. The special partitions are named PAGE, BOS, LOG, SALV, and DUMP. (There is no longer any need for an explicit MULT partition; all storage described on VOL cards makes up the storage used by Multics.) The PRPH card specifies that the peripheral known as DSK is a model 191 disk. The physical volumes are named V1, V2, and V3; there are two logical volumes, ROOT and USER.

There have been several simplifications made in the scheme proposed here over the plan outlined in MTB-110. By placing the partitions inside physical volumes, we eliminate the bothersome "mini-pack" implementation, with its attendant threats of multiple labels per pack and the necessity of relocating record addresses.

The method of describing the paging device and the PAGE partition shown above may be overcomplicated. Actually, PAGE need not be a partition known to BOS, except for the times when the operator wishes to use the TEST command to test addresses on the device or to clear the paging device.

OTHER REMARKS

In order to simplify some code in page_fault, the paging device will be assigned a PVT index if it is present in the configuration.

When the operator types an address to BOS, he may say one of

```
PART <partid>
VOL <name> <recordno>
PHY <iom> <channel> <drive> <sector>
```

such addresses will be used in the SAVE, RESTOR, and TEST commands. The physical-mode addresses, beginning with PHY, are the only way the operator can specify the sectors not part of any record.

The COLD and WARM cards for BOS bootload must be modified slightly also. The new format is:

```
COLD <iom> <channel> <drive> <frec> <nrec>
```

where frec and nrec are regular Multics record numbers. BOS secondary storage allocation will be changed so that all BOS storage lies inside records addressable by Multics (that is, we will not use the 512 words per cylinder which is not part of any record). This change should permit an extension of the storage system which will cause each special partition to be described by a segment or multi-segment file in the directory >parts; these branches will be constructed or adjusted at bootload time. Manipulation of the SYSERR log, copying of FDUMP's, and even installing new versions of BOS become vastly simpler with such an arrangement.

The CONFIG command of BOS will check the PART BOS card against the values specified on the COLD or WARM card and complain if there is any discrepancy. If no PART BOS card is found, one will be generated. However, BOS will not attempt to access the label of the BOS pack at load time.

APPENDIX - New Data Structures

The following declarations describe the current plans for data layout of disk storage for the new storage system.

Readers are cautioned that these declarations are tentative and may change at any moment.

Declaration of Volume Label

```
dcl 1 label based aligned,
    2 version fixed bin,
    2 mfg_serial char (32),
    2 pv_name char (32),
    2 pv_id bit (36),
    2 lv_name char (32),
    2 lv_id bit (36),
    2 root_pv_id bit (36),
    2 time_registered fixed bin (71),
    2 nb_pv_in_lv fixed bin,
    2 vol_size fixed bin,
    2 vtoc_size fixed bin,
    2 pad1 (43) fixed bin,
    2 time_mounted fixed bin (71),
    2 time_map_updated fixed bin (71),
    2 time_unmounted fixed bin (71),
    2 time_salvaged fixed bin (71),
    2 n_bad_records fixed bin,
    2 err_hist_size fixed bin,
    2 pad2 (54) fixed bin,
    2 root,
      3 here bit (1),
      3 root_vtocx fixed bin (35),
      3 shutdown_state fixed bin,
    2 pad3 (60) fixed bin,
    2 nparts fixed bin,
    2 parts (48),
      3 name char (4),
      3 frec fixed bin,
      3 nrec fixed bin,
      3 pad5 fixed bin,
    2 pad4 (10*64) fixed bin;
```

version is the version number. This version is 1.

mfg_serial	is the manufacturer's serial number
pv_name	is the physical volume name.
pv_id	is the unique ID of this volume
lv_name	is the name of the logical volume containing this physical volume.
lv_id	is the unique ID of this physical volume's logical volume.
root_pv_id	is the unique ID of the volume containing the root. all volumes must agree.
time_registered	is the time the volume was registered by the system
nb_pv_in_lv	is the number of physical volumes in the logical volume.
vol_size	is the total size of the physical volume, in records.
vtoc_size	is the total size of the overhead region of the pack, including the 8-record fixed area and the records used for the VTOC.
pad1	is padding
time_mounted	is the last time the volume was mounted
time_map_updated	is the last time the volume map was known good
time_unmounted	is the last time the volume was unmounted cleanly
time_salvaged	is the last time the volume was salvaged
n_bad_records	is the number of unusable records on the volume.
err_hist_size	is the size of the volume error history, in records
pad2	is padding
root.here	is TRUE if the root is on this pack

root.root_vtocx	is the VTOC index of root, if it is here
root.shutdown_state	is the status of the storage hierarchy. This variable is set to indicate whether the salvager has been run, and whether it successfully repaired damage to the system. It is inspected by the BOS command IF to control the flow of RUNCOMs.
pad3	is padding
nparts	is the number of special partitions in the label.
parts.name	is the name of a special partition.
parts.frec	is the first record address for the special partition.
parts.nrec	is the number of records in the special partition.
parts.pad5	is padding
pad4	is padding

Declaration of VTOC_Header

```
dcl 1 vtoc_header based aligned,
    2 version fixed bin (17),
    2 n_vtoce fixed bin (17),
    2 vtoc_last_recno fixed bin (17),
    2 n_free_vtoce fixed bin (17),
    2 first_free_vtocx fixed bin (17),
    2 pad (1024-5) bit (36);
```

version	is the version number. The current version number is 1.
n_vtoce	is the number of vtoc entries
vtoc_last_recno	is the record number of the last record of the vtoc
n_free_vtoce	is the number of free vtoc entries
first_free_vtocx	is the index of the first vtoce in the free list

pad is padding

Declaration_of_Volume_Map

```
dcl 1 vol_map based aligned,
    2 n_rec fixed bin (17),
    2 base_add fixed bin (17),
    2 n_free_rec fixed bin (17),
    2 bit_map_n_words fixed bin (17),
    2 pad (60) bit (36),
    2 bit_map (3*1024 - 64) bit (36);
```

n_rec	is the number of records represented in the map.
base_add	is the record number for the first bit in the bit map.
n_free_rec	is the number of free records.
bit_map_n_words	is the number of words of the bit map.
pad	is padding.
bit_map	is the bit map; the entire volume map occupies 3 records.

(END)